

AD-A240 432



ON PAGE

Form Approved
OMB No. 0704-0188Public reporting burden
maintaining the data
suggestions for red
and to the Office ofresponse, including the time for reviewing instructions, searching existing data sources, gathering and
comments regarding this burden estimate or any other aspect of this collection of information, including
ation Operations and Reports 1215 Jefferson Davis Highway Suite 1204, Arlington, VA 22202-4302
Arlington DC 20503

1 AGENCY USE ONLY (Leave blank)

2 REPORT DATE

August 1991

3 REPORT TYPE AND DATES COVERED

professional paper

4 TITLE AND SUBTITLE

DATABASE MANAGEMENT SYSTEM INTERFACE STANDARD FOR NAVY
NEXT GENERATION COMPUTING RESOURCES

5 FUNDING NUMBERS

PR: CC30
PE: OMN
WU: DN587574

6 AUTHOR(S)

D. L. Small, M. C. Butterbrodt, R. M. Bergman

7 PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Ocean Systems Center
San Diego, CA 92152-50008 PERFORMING ORGANIZATION
REPORT NUMBER

9 SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space and Naval Warfare Systems Command
Washington, DC 20363-510010 SPONSORING/MONITORING
AGENCY REPORT NUMBER

11 SUPPLEMENTARY NOTES

12a DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b DISTRIBUTION CODE

13 ABSTRACT (Maximum 200 words)

This paper is a "snapshot" of the Navy's use of database management system (DBMS) technology for Command, Control, and Combat Systems. This paper traces similarities in DBMS interface needs throughout current and envisioned Navy systems.

91-10548

Published in *Proceedings of Database '91 Merging Policy, Standards, and Technology*, 1991.

14 SUBJECT TERMS

command control
computer sciencessoftware tools
support software
Ada

15 NUMBER OF PAGES

16 PRICE CODE

17 SECURITY CLASSIFICATION
OF REPORT

UNCLASSIFIED

18 SECURITY CLASSIFICATION
OF THIS PAGE

UNCLASSIFIED

19 SECURITY CLASSIFICATION
OF ABSTRACT

UNCLASSIFIED

20 LIMITATION OF ABSTRACT

UNCLASSIFIED

<div>21a NAME OF RESPONSIBLE INDIVIDUAL D. L. Small</div>	<div>21b TELEPHONE (include Area Code) (619) 553-4130</div>	<div>21c OFFICE SYMBOL Code 413</div>
---	---	---

Paper presented at Database '91 Merging Policy, Standards
and Technology on June 25, 1991

White Paper on the
Database Management System Interface Standard
for Navy Next Generation Computing Resources



Naval Ocean Systems Center
271 Catalina Boulevard
San Diego, California 92152-5000

Accession For	
DTIC GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Dana L. Small, Code 413
Mary C. Butterbrodt, Code 413
Richard M. Bergman, Code 412

May 1991

Final Revision #2

CONTENTS

1	INTRODUCTION	1
2	BACKGROUND ON NAVY SYSTEMS	2
	2.1 TACTICAL SYSTEMS	3
	2.2 STRATEGIC SYSTEMS	3
	2.3 REAL/CRITICAL TIME PERFORMANCE	4
	2.4 DATA CONSISTENCY	4
	2.5 HETEROGENEOUS	5
	2.6 FAULT TOLERANCE/RELIABILITY	6
	2.7 DBMS LOGISTICS SUPPORT	7
3	INTERFACE NEEDS	7
4	MULTILEVEL SECURITY	8
5	SUMMARY	9
6	REFERENCES	10

①

1 INTRODUCTION

This paper is a "snapshot" of the Navy's use of database management system (DBMS) technology for Command, Control and Combat Systems. Our long-term goal is to contribute to the development of a standard DBMS interface (DBMSIF) to promote interoperability among Navy systems. This paper is an excerpt from the paper that was developed under Naval Ocean Systems Center (NOSC) tasking for the SPAWAR 3243, Next Generation Computer Resources (NGCR) Program.

Navy systems have a requirement for managing a massive command, control, communications, and intelligence (C³I) system encompassing land, surface, subsurface, air, and space data elements. These systems ultimately control thousands of complex sensor, combat direction and weapon systems aboard hundreds of tactical units. Driving such systems are significant requirements for management of such objects, discriminating the real threats among them, and tracking them with realtime updates using an intelligent analysis of which objects are benign (friendly, neutral or decoys) and which are threats. The systems are necessarily distributed and require substantial data which must be consistent through time, often requiring the meeting of a hard realtime deadline schedule for data availability and accessibility. To succeed a thorough, consistent, and logical data model must be used for all dispersed components of the Navy's (C³I) and combat systems. The model must be based on multiple disparate large databases, all of which require timely, consistent, and uniform access.

The Navy requirements for DBMS and the standards for its use have been, at least, in the tactical and strategic areas, very informal to almost ad hoc; that is to say, very project-requirements oriented. Two universal reasons for this are performance (very slow and cumbersome), and memory (high memory budgets for both internal memory of computers and external storage). A third reason is that there has been a lack of formal operating systems or operating system interface for a DBMS to "hook to" in such systems. Tactical weapon systems require stringent performance within the realtime to critical realtime performance envelopes. In such cases there are hard deadlines to meet with only a finite amount of processing time available. What follows is a simple example of this: There are a number of inbound identified hostile targets on a task group where the number of weapons to be assigned to the closest hostile targets isn't quite enough. If the computer system performance is critical-time oriented in nature, the assigned weapons could be

deployed, the next available weapons loaded and replaced, the next iteration of computation completed, and that volley of weapons deployed, all in performance responsive to the mission requirements for task group defense. Another example could be responding to the identification of low observable aircraft and enabling interception before they reach their targets or go beyond range. Most of the war-fighting "computer code" running in Navy systems today use a significant level of hand tailored assembly-level code to meet this type of threat performance requirement rather than using a set of standardized interface tools for management of data and resource scheduling. Such code also does not provide for interface between responsive target management and identification of likely target point of origin where such data likely resides in a large "unresponsive" database system.

This paper traces similarities in DBMS interface needs throughout current and envisioned Navy systems.

2 BACKGROUND ON NAVY SYSTEMS

In the strictest interpretation of NCR development of standards, the DBMS standard shall be an INTERFACE standard. The objectives for standardizing a DBMSIF are to promote application system portability, interoperability, software maintenance and reuse, as well as a more common and meaningful representation of data throughout a system.

Most of the current tactical weapons and sensor systems deployed today (especially the systems utilizing the Navy standard computers and CMS-2 and other pre-Ada languages) do "data management", but not with formal DBMS structures. These software structures are, for the most part, handwritten and tailored around very primitive and fundamental "executives" and "kernels" used as operating systems. These are used for two main reasons: (1) DBMS structures did not exist as part of the Navy standard software products at the time most of these systems were designed and built (and still don't for the languages used for these systems) and (2) performance in critical time (hard deadline) situations is of primary concern. Only within the last year has Ada started to provide within its program library the functionality of the SQL language. The DBMS issue is still approached as application software which will meet the requirements of the mission for which the system is being designed and built. An exception is the LHA amphibious ships' use of the Management Information System (MIS) as a general purpose data storage, processing and retrieval system with application to Navy administrative, tactical and strategic operations. The hierarchical database structure was used for the MIS on the AN/UYK-7 computer system because of a "non-measured" belief as to its performance. The MIS system, for example, is used for setting up the planning for amphibious operations. After

deployment, independent processing with the same computer system can be done using the Tactical Data System (TDS) with all track flat files maintained in memory. Note here MIS planning is carried out primarily before "time critical" operations take place.

2.1 TACTICAL SYSTEMS

Tactical system target information has routinely been supported by linear flat files of tracks identified formally in message format sent between ships. With the advent of the Advanced Combat Direction System (ACDS), it has become apparent that a linear file with a minimal number of attributes is not adequate for today's track file management. An Object-Oriented (OO) data system is being used for threat management for ACDS to assist in the manipulation of some of these extra attributes. But there again DBMS is not used for a major portion of the CDS.

2.2 STRATEGIC SYSTEMS

DBMS is used for the Navy's strategic systems. The Navy World Wide Military Command Control System (WWMCCS) Standard System (NWSS) started by using the network database model to manage the various files for which CINCLANTFLT and CINCPACFLT have responsibility (e.g., FORSTAT, MOVEREP, equipment capabilities, and personnel information). In the mid 70's experiments were performed to move these files into a relational database format (due primarily to the Advanced Command Control Architectural Testbed [ACCAT] program). Due to the success of those experiments, the Navy is now beginning to convert present non-relational Navy systems over to relational database systems. Because of the conventions followed in network database systems, early conversion to relational systems used network links as column entries in relations--an easy technique to follow, but with performance degradation as a result. It is just within the last few years that the relational system design is being reconsidered. The technique of indexed columns--a non-SQL standard--is an extension used to improve performance by allowing access to only a few columns, instead of the large number considered for the description of a relation entity. The cost is a larger memory budget. A major example of such a conversion is the Naval Warfare Tactical Database (NWTDB) being used at NWSS. All such sites will be using the M204 database system built by Computer Corporation of America (CCA). The M204 database system is an IBM hierarchical data model which is being converted over to a relational DBMS based on the SQL standard.

The Navy, in its Naval Tactical Command System-Afloat (NTCS-A) program, is intending to use not only NWTDB and some other smaller static relational systems but also a static system called

Military Intelligence Integrated Database System (MIIDS). Within NTCS-A, these strategic static databases can be combined with the more tactical realtime Afloat Correlation System (ACS) and ACDS. Here, as is the case for ACDS, there is a significant issue of management of realtime data consistency while querying large databases (upwards of gigabytes in size in raw format for MIIDS data). For ACDS identification data, and NWSS, MIIDS and NWTDB data there is the issue of how to design the DBMS system so that retrieval performance does not suffer. It does not necessarily follow that techniques used in older systems, i.e., using the network database model as available in WWMCCS/NWSS, and linear files as used for processing of NTDS/ACDS tracks, will allow for good performance for database access for database systems with a much greater volume of data. These issues require detailed analysis to develop a versatile, adaptable interface standard.

The same type of issues, noted above for strategic systems, are being faced in the Navy's Intelligence Systems. In these cases, more data has to be handled, much of which is more free-form, textual and static in nature.

2.3 REAL/CRITICAL TIME PERFORMANCE

For real/critical time data, the issues of maintenance and management of realtime data consistency (data is not lost) while querying large data bases (upwards of gigabytes in size in raw data) is tantamount, e.g., the ACDS and NCTS-A efforts referred to previously in this paper. This is especially difficult where processing of simultaneous external asynchronous events is required. In order to ensure that data are processed in realtime, memory and other critical resources can be allocated and scheduled in close cooperation with the operating system. If memory is appropriately scheduled anywhere on the distributed database network, a transaction could run in realtime. From the realtime scheduling point of view, the primary problem introduced by sharing distributed data is the blocking caused by the locking (or time stamp) protocols for concurrency control, which often cause unacceptable delays. However, concurrency control protocols are necessary to ensure the consistency, integrity and predictability of the shared data (database) and the correctness of distributed computations (transactions). Experimentation is being performed at NOSC (Butterbrodt and Green 1990) analyzing the performance of such systems.

2.4 DATA CONSISTENCY

For static Navy (C³I) data there is the issue of how to design the database system so that retrieval and execution performance do not suffer, yet consistent data is provided. The technique of indexed columns--a non SQL standard for relational database

systems--is an extension that can be used to improve performance by allowing access to only a few columns of a relation. But that is an adhoc way to improve database query response. Any application requiring non-realtime (NRT) large databases with few updates and/or realtime (RT) data for threat analysis and weapons deployment requires timely and consistent access to that data (See the previous reference to the LHA DBMS).

Such designs must assure that data upon their arrival will be available, retain referential consistency with other data copies, and not be mistakenly lost during updates or access of large data bases. Potential for concurrency of processing must be available for determining location and accessibility of the data wherever they reside. The designs must provide responsive data access and good potential for decision quality in access and identification of data. They must allow for dynamic re-allocation of relation locations based on processor availability.

A study is currently underway at NOSC (Small 1990) presenting design options for a realtime distributed database system in support of Naval Tactical Command System Afloat (NTCS-A) databases. It provides support to ensure that data upon its arrival will be accessible and consistent with other available copies. The design consists of a directory available at each node in the distributed system which can be accessed by any application requiring realtime and non-realtime data for threat analysis and warfare planning. All accesses will be provided by SQL relational syntax commands.

2.5 HETEROGENEOUS

The DBMS interface standard must be implementable on a wide variety of hardware architectures, configurations, and capabilities, because more than one methodology and vendor's DBMS may be involved (note the success of ORACLE relational DBMS on a variety of different computer platforms). This pertains not only to the DBMS resident on a single processor, but also to a DBMS which spans multiple heterogeneous processors. The automatic exchange of information between heterogeneous and multimedia (text, graphics, images and sound) is also an issue. Software requirements can include access to flat files, network, hierarchical, and object-oriented databases as well as relational databases, many of which are very large. These systems can be based on the Navy's AN/UYK computer systems, SUN Microprocessors, VAXes, PC's, embedded processors, e.g., 68030 boards and parallel processing machines such as the ENCORE processing system or the Enhanced Modular Signal Processor, AN/UYK-2 (EMSP). This explanation should not preclude the possibility that some parts of the DBMSIF could actually be implemented in hardware itself to satisfy extreme performance requirements.

One should also note that all query languages using relational DBMSs are not alike. Most vendor's SQL implementations vary in SQL support (built-in functions, relational operators), SQL syntax, SQL semantics (return codes, data type handling), transaction handling (commit processing, concurrency control, data isolation options), and data dictionary format. To date, there are no solutions to this problem. However, there is a clear industry trend towards both heterogeneity and distribution.

DISTRIBUTED SYSTEMS

Distributed systems will continue to gain importance and application within the Navy into the 1990's and beyond. The data management issues in distributed systems will clearly be among the more difficult issues to resolve. The data types and volume of data, data currency, data consistency among parallel processes, along with the data fusion issues, stress the need for a DBMS interface standard for future success in Navy NGCR systems. The standard must support a vertical and horizontal hierarchy across systems, scaleable from the simplest to the most complex Navy systems, and still provide required performance. The primary objective of a distributed DBMS is to give interactive query users and application programs transparent access to remote data as well as local data. If, for example, there is a track file (T) located in the Naval Tactical Data System (NTDS) and the Advanced Combat Direction System (ACDS), and a ships file (S) located in the Naval Warfare Tactical Database (NWTDB), a distributed database (DDB) would allow a user located anywhere on the network to physically or logically enter a SQL statement to access data from the T and S files. The methods used to access remote data could conform with the emerging Open Systems Interconnection (OSI) standard (Mollet 1990) or it could conform to SAFENET's (Survivable Adaptable Fiber Optic Embedded Network) lightweight protocol suite for realtime applications.

2.6 FAULT TOLERANCE/RELIABILITY

This is an increasingly important area where DBMSs are expected to provide support. Fault tolerance and reconfiguration are of prime concern in Navy tactical systems for mission effectiveness. Redundancy and multiple points of access/connection can be major considerations for such systems. If parts of a system receive damage from either accident or combat, it is imperative that the system maintain an operating status to support the particular mission. Data currency, consistency, and completeness are as or more important than the communication paths which are used for access. Optimal update strategies are needed when confronted with such circumstances.

The interface standard must support the capability to be creative in the above mentioned areas in distributed, heterogeneous systems when designing for fault tolerant, dynamically reconfigurable modes of operation. One of the most important areas of resource management is the integrity of the data necessary to perform the mission. Data loss, data quality and data accessibility are the essence of recovery from interrupted service or operation in degraded modes of a system. The interface standard and subsequent products developed against the standard must be sensitive to these issues.

Standardization efforts in these areas are unknown, and any work done here is application and requirement specific. The specific areas of "commit" and "rollback" should be studied more for incorporation into systems for more reliability and integrity of data kept between and among inter- and intra-databases. This concept may take a toll on performance and is as much a network and operating system issue as it is a DBMS issue. The research on DATA CONSISTENCY (Small 90) discussed above could have an impact if integrated into a total concept for fault tolerance.

2.7 DBMS LOGISTICS SUPPORT

A major issue is preservation of data, even though the database system being used may change or new functions (in the "data maintenance" mode) made to work on the data. Referential integrity maintenance is key to such preservation, just as it is for fault tolerance and recovery and multilevel security. In providing such changes, there must be assurance of non-contamination of data to be entered in the database system and that data do not become lost or stale (out-of-date).

3 INTERFACE NEEDS

Based on the requirements of consistent realtime, distributed, and heterogeneous systems, there are a number of interface needs that must be established. The DBMSIF standard should provide DBMS interface independence whether the data model requires use of flat files, or hierarchical, network, relational, or object-oriented data.

The DBMS interface must be defined in a way that is independent of any particular programming language. It must be possible to access the services of the DBMS by Ada as well as other language applications in common use in Navy Systems, such as C, COBOL (Common Business Oriented Programming Language), FORTRAN (Formula Translation Programming Language), CMS-2 (Compiler Monitor System, Version 2), the Navy's standard programming language prior to Ada, artificial intelligence languages, natural language front ends, and signal processing-type languages.

The DBMS interface standard should interoperate with the POSIX standard for the operating system interface (POSIX 1003/UniForum, 1990/NGCR OSSWG Report). P1003.1 defines the interface between portable application programs and the operating system, and supports application portability at the source code level. This operating system standard interface will allow programs to be written for a target environment in which they can be ported to a variety of systems. A DBMS should be able to request resource management from interfacing applications such as operating systems.

Ideally, there should be an interface with a wide variety of network architectures which is transparent to the DBMS interface. The network interface might conform to POSIX 1003.8-Network Services (UniForum, 1989). Such a use could permit transparent sharing of distributed files across systems. The DBMSIF should support media- and protocol-independent applications, and be consistent with existing and emerging standards such as Open Systems Interconnection (OSI). It should work with local area networks (e.g., SAFENET 1990) and wide area networks. It should interface with a variety of different network architectures (e.g., OSI, SAFENET, Transmission Control Protocol/Internet Protocol [TCP/IP], Systems Network Architecture [SNA], Digital Equipment Corporation Network [DECNET]).

4 MULTILEVEL SECURITY

Like realtime, security traditionally has not been of as great concern to industry as to the military. The military is concerned about issues of confidentiality, data integrity, non-repudiation, proof of origin and submission for all the systems described above. But also like realtime, these security issues are now becoming increasingly important to systems in use for nuclear control and banking. Security implications are not well understood in such systems, but there are guidelines and the requirement is real. Such a system should ensure that users cleared at different security levels can access and share a database without violation of security. Industry's recognition of this is demonstrated by yet another POSIX subgroup, P1003.6, working on security enhancements interface (POSIX 1003.6/UniForum, 1990/NGCR OSSWG Report).

There are two installed, procurable database systems that we know of that have been certified and accredited to run controlled/limited multilevel security. These are the M204 system now being installed as the WWMCCS standard system, and the Honeywell Integrated Data Store 2 that has formed the basis for the Navy's World Wide Data Management System, WWDMS and WWMCCS. None of these provide an open client/server system. In all instances, the system requires specialized machine and/or operating system support.

Referential integrity, i.e., maintenance of consistency when changing related data in other tables, must be provided whenever SQL commands such as insert, delete, and update are used. The combination of access control with management of such data integrity using data validation and consistency constraints can provide a reasonable level of security for user access to data. This does not say that the system is tamper proof or has met all requirements for a secure system. Providing such access control does provide a starting place for multilevel security. Multilevel secure relational systems are now under development to be able to give different views of data at different security levels. The mechanisms use polyinstantiation to allow two different views of tuples with the same primary key to exist. Maintenance and management of items in such a system can be costly in performance, particularly if that must be met for each row of a table or record of a file. But regardless of such potential cost, new systems under development which operate in a multilevel secure environment (whether they are object oriented, knowledge base, realtime, distributed database systems, etc.) must provide for such security at the beginning of their development. There is currently under way research in inference and aggregation using multilevel secure systems, and SQL extensions, integrity policies and automated auditing techniques, all to support more secure systems.

As "trusted operating system" technology matures, it is foreseeable that the operating system will handle the majority of the security issues as systems resource issues and functions before the data is received by the database. Due to timing and performance considerations, as well as from a security point of view, the database must interface directly with memory management software, device driver software and the system scheduler. From the "Orange Book" (NCSC, 1990) Trusted Computing Base (TCB) point of view, the DBMSIF could interface with a security reference monitor which could provide a subset M of the TCB. It will exist below the interface standard transparent to applications. M must be tamper resistant and small enough to be subject to analysis and testing. The "completeness" of M must be assured. From the context of the NGCR OS, a subset of POSIX could provide the M (POSIX 1006.3/UniForum, 1990/NGCR OSSWG Report). The access control policies (P) of POSIX's M must provide at least:

- 1) a guarantee that there is NO violation or penetration of protected memory space by unauthorized users (including logon guarantee from the OS point of view), and
- 2) from the multitasking point of view, a guarantee that there is no mix-up between tasks, and no invasion of code or memory used between tasks.

5 SUMMARY

All of the database applications described in this paper that the Navy is using or planning to use require at least the following provided or supported across standard interfaces between database and other applications:

1. Maintenance of consistent data whether data is being updated in realtime or not.
2. Ability to make notification of significant data changes which could happen in non-realtime data.
3. Requirement, in all cases, for maintenance of referential integrity.
4. Control, in all instances, of any or all system scheduling and resource utilization, i.e., memory, network, device driver software.

In each case, the interface described can be provided at the operating system level of control.

6 REFERENCES

Butterbrodt, M.C., and J.C.M. Green, 1990. "The Distributed Operating System Experiment: A Vital Link in the Development of a Realtime Relational Database Environment", CDRL Item produced. Naval Ocean Systems Center, San Diego, CA. Available upon request.

Mollet, A. A., Computer Sciences Corporation, 1990. "Distributed Query Processor Research", NOSC Technical Document 2053 produced for NOSC Code 413, San Diego, CA., 92152-5000

NCSC 1990, third draft of a Trusted Database Interpretation (TDI) for its Orange Book, officially the Trusted Computer Security Evaluation Criteria, October 1990.

NGCR OSSWG Recommendation Report, May 1990

SAFENET TADSTAND B, Revision B, June 1990

Small, D. 1990. "Distributed Realtime Database System Navy Command/Control Systems - Afloat (NCCS-A) Project, NOSC TD 1798 (April 1990). Naval Ocean Systems Center, San Diego, CA., 92152-5000

UniForum, 1989. Your Guide to POSIX.

Vugraphs used at the presentation the evening of June 25 follow:

NUMBER 1

White Paper on the
Database Management System Interface [DBMSIF] Standard
for Navy Next Generation Computing Resources

Naval Ocean Systems Center
271 Catalina Boulevard
San Diego, California 92152-5000

Dana L. Small, Code 413
Mary C. Butterbrodt, Code 413
Richard M. Bergman, Code 412

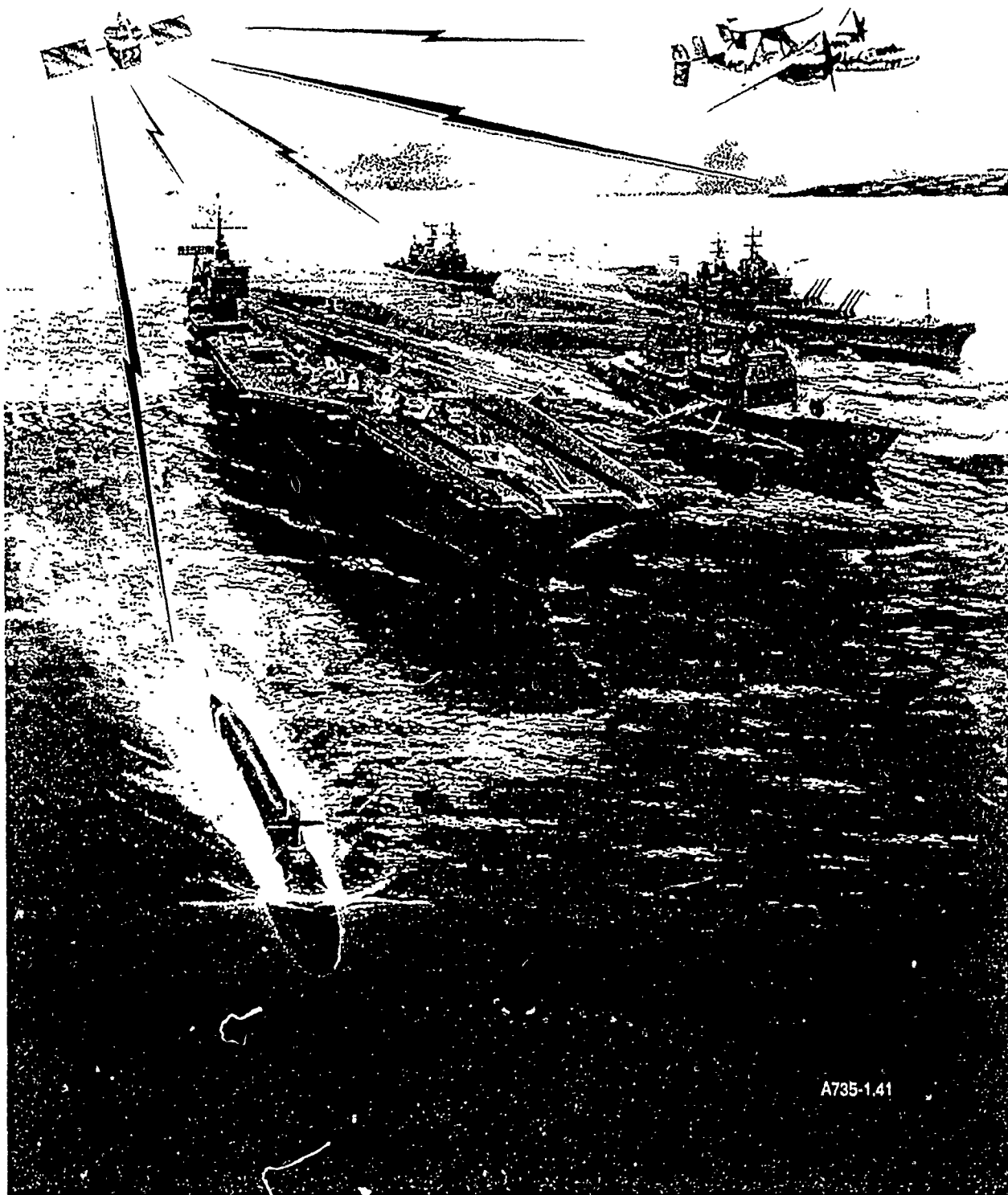
NUMBER 2

OPEN SYSTEMS ENGINEERING ARCHITECTURE

- . PROVIDES FRAMEWORK FOR SYSTEMS DESIGN
 - DOES NOT DEFINE OR STANDARDIZE ON A COMPUTER DESIGN
 - STANDARDIZES HARDWARE / SOFTWARE INTERFACES
 - PROVIDES FRAMEWORK FOR INDUSTRY IR&D INVESTMENT
- . REMAINS CURRENT WITH STANDARDIZATION TRENDS IN COMMERCIAL SECTOR
- . BEING IMPLEMENTED THROUGH JOINT NAVY/INDUSTRY WORKING GROUP
 - WIDELY USED NON-PROPRIETARY COMMERCIAL STANDARDS BASE
- . THIS PAPER'S EMPHASIS
 - DATABASE MANAGEMENT SYSTEM INTERFACE (DBMSIF) STANDARD

NUMBER 3

NAVY USE OF DATABASE MANAGEMENT SYSTEMS (DBMS')
REQUIRES MULTIPLE LARGE DISPARATE DATABASES



A735-1.41

Navy (C³I) systems.

CONTAINING COMMON INFORMATION

WITH TIMELY, CONSISTENT AND UNIFORM ACCESS

NUMBER 4

IN SUCH AN ENVIRONMENT HARD DEADLINES MUST BE MET WITH
ONLY FINITE AMOUNT OF PROCESSING TIME AVAILABLE
FOR EXAMPLE:

THREATS BEING FACED BY CARRIER BATTLE GROUP
WITH SATURATION RAID OF TACTICAL AIRCRAFT

WHICH COULD BE COMPOUNDED BY STANDOFF JAMMERS
DEGRADING SENSOR PERFORMANCE

AND/OR CRUISE MISSILES BEING FIRED FROM DIFFERENT
TYPES OF PLATFORMS

IN ALL SUCH CASES, OPTIMAL COMPUTER SYSTEMS PERFORMANCE
AND INTEROPERABILITY REQUIRED FOR ENGAGEMENT
RESPONSIVENESS FOR DETECTION, CLASSIFICATION AND
SCHEDULING

NUMBER 5

PICTURE

NUMBER 6

TODAY'S USE OF DBMS'

NO OPERATING SYSTEM INTERFACE TO HOOK INTO

LACK OF PERFORMANCE BY DBMS'

HIGH MEMORY BUDGETS REQUIRED

RESULTING IN SIGNIFICANT LEVELS OF HAND TAILORED
ASSEMBLY LEVEL CODE

NUMBER 7

NAVY DBMS INTERFACE NEEDS

INDEPENDENCE REQUIRED

WHETHER DATA MODEL USES FLAT FILES, HIERARCHIAL,
NETWORK, RELATIONAL OR OBJECT-ORIENTED DBMS'

NO MATTER WHAT TYPE OF NETWORK ARCHITECTURE IS USED

DBMS ACCESS MUST BE INDEPENDENT OF ANY PARTICULAR
PROGRAMMING LANGUAGE

ALL SUCH ACCESSES ARE TO SUPPORT ADA BINDINGS

NUMBER 8

INTERFACE NEEDS CONT'D

DBMS MUST INTERFACE WITH THE OPERATING SYSTEM
SELECTED FOR NGCR, POSIX

POSIX WILL CONTROL SYSTEM SCHEDULING

RESOURCE UTILIZATION, INCLUDING

MEMORY, DEVICE DRIVERS, NETWORK

TO ENSURE ALL PROCESSING DEADLINES ARE MET

REQUIREMENTS FOR MULTILEVEL SECURITY MUST BE
FACTORED IN INCLUDING AT LEAST:

MAINTENANCE OF INTEGRITY OF DATA BY PROVIDING
DATA VALIDATION AND CONSISTENCY CONSTRAINTS

NUMBER 9

SUMMARY

ALL OF THE NAVY'S DBMS APPLICATIONS REQUIRE AT LEAST
THE FOLLOWING:

MAINTENANCE OF CONSISTENT DATA
WHETHER THEY ARE UPDATED IN REALTIME OR NOT
AND MEET ALL PROCESSING DEADLINES

NOTIFICATION OF SIGNIFICANT DATA CHANGES
WHEREVER THEY ARE ACCESSED

OS CONTROL OF SYSTEM SCHEDULING AND RESOURCE UTILIZATION
REALTIME OR NOT

UNIFORM ACCESS TO DATA VIA DBMS QUERY
AND/OR COMPUTER PROGRAMS